REMARKS

In a final office action dated March 29, 20043, the Examiner rejected claims 1-2 and 20-21 under 35 U.S.C. §103(a) as obvious over Ebrahim et al.(US Patent 5,905,998). Claim 10-15, 17, 19, 29-34 and 38 were allowed.

Independent claims 1 and 20 have been amended to clarify certain matters, if such clarification be necessary.[1] In particular, claims 1 and 20 have been amended to recite what happens if a command requesting invalidation of a cache line completes normally, i.e., the castback command is discarded, and no action is taken. Claim 1 has additionally been amended to explicitly recite that the snooping step takes place after invalidation, i.e., that the processor receiving an invalidation request invalidates the data without waiting for a system response to the command. This was already explicit in claim 20. As amended, the claims are patentable over the cited art.

In accordance with one aspect of applicants' invention, when a certain type of invalidation command (e.g., a kill command) requesting invalidation of a cache line is received at a processor, and the processor's cache has such a cache line containing modified data, a cast back command is temporarily put in the transition cache. The transition cache then waits for the system response to the invalidation command. If the system response is normal completion, then the data must be

[1] Applicants do not believe the claim amendments are necessary to distinguish over *Ebrahim*, for reasons stated in response to the previous office action. However, in the final office action, the Examiner, in response to applicants' arguments, appears to indicate that the lack of an explicit claim recitation of what happens if the invalidation command completes normally is a significant problem with the claims. Of course, in this case, nothing happens, the data having already been invalidated. Although this appears to be an unnecessary recitation, applicants are willing to add it if it will further prosecution of the present application.

discarded to maintain coherency. But if the system response is a retry, then it is possible to cast back the modified cache line to main memory.

Ebrahim discloses only that, in similar circumstances, the cache line is discarded (i.e, the first part of the response). *Ebrahim* does not disclose any capability to cast back the modified cache line to memory if the invalidation command does not complete.

The Examiner's argument appears to be that *Ebrahim*'s disclosure that "... the writeback can be canceled if the same data block is invalided by another data processor..." (*Ebrahim*, col. 4, lines 22-24) is equivalent to saying "the write back can be cancelled only if the invalidation command completes (Office Action, p. 7). It appears to applicants that the Examiner is confusing the difference between the act of ***invalidating*** a cache line in the processor and the ***completion of a command requesting invalidation*** of a cache line, from the perspective of the system. This is understandable, since *Ebrahim* does not make a clear distinction between these two events. And why should it? Under conventional art such as *Ebrahim*, there is no capability, as disclosed by applicants, to wait for the system response and conditionally castback the data, depending on the response. It is therefore useful to review the sequence of events clearly. The following events occur sequentially, in the order indicated:

(1) a first processor issues a command requesting invalidation of data in the caches of other processors;

(2) a second processor snoops the command, and responsive thereto, invalidates its data; and

(3) a system response to the command is generated; this system response may be normal completion, or may be retry, ***but in either case step (2) above has already been performed.***

When *Ebrahim* states at col. 4, lines 18-25, that "... the writeback can be canceled if the same data block is invalided by another data processor...", *Ebrahim* is referring to step (2) above, not step (3). Invalidation of the data occurs in response to receipt of a command requesting invalidation, and for this purpose, it is irrelevant whether or not the command completes normally.

In order to clarify this point, if such clarification be necessary, applicants have amended claim 1 to recite that the snooping occurs after invalidation. Applicants have further amended claims 1 and 20 to recite the converse of the proposition that, if the system response is a retry, a castback is issued on the bus from the transition cache. Specifically, the converse of this proposition is that, if the system response is not a retry (normal completion), a castback is not issued on the bus (the command is discarded). Applicants do not believe that these amendments are strictly necessary to distinguish over the art, but in the interests of furthering prosecution and/or narrowing issues for appeal, applicants are willing to make the amendments clarifying these points.

Far from suggesting applicants' claimed invention, *Ebrahim* teaches away from the invention on the very point of novelty which distinguishes over the cited art. *Ebrahim* states that a dirty victim is normally written back, except that the write back is canceled if the same data block is invalidated before the writeback becomes effective. The point of applicants' invention is that the write back ***is not always cancelled*** if the same data block is invalidated before the writeback becomes effective. Invalidation, as explained herein, occurs responsive to receiving the command requesting invalidation, but in some circumstances (specifically, when the system response to the command is a retry), the writeback or castback occurs, notwithstanding that the data has already been invalidated. *Ebrahim* does not teach or suggest such a feature. Rather than suggest this capability, the sentence quoted by the Examiner teaches away from it by stating

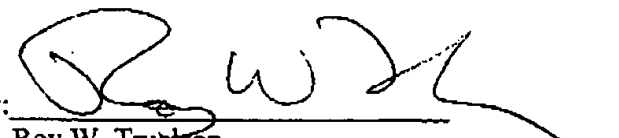
"except that...the write back can be canceled if the same data block is invalidated..." The clear implication is that, in these circumstances, no write back occurs.

Finally, *Ebrahim* does not disclose any mechanism whereby modified and invalidated data can be stored in a transition cache pending a determination whether the invalidation command actually completed normally. The lack of any such mechanism is further evidence that *Ebrahim* does not teach or disclose the claimed feature.

For all of these reasons, amended claims 1 and 20 are neither taught by, nor obvious over, *Ebrahim*, and are patentable. Claims 2 and 21 are dependent on claims 1 and 20, and likewise patentable.

In view of the foregoing, applicants submit that the claims are now in condition for allowance and respectfully request reconsideration and allowance of all claims. In addition, the Examiner is encouraged to contact applicants' attorney by telephone if there are outstanding issues left to be resolved to place this case in condition for allowance.

Respectfully submitted,
DONALD L. FREERKSEN, et al.

By: 
Roy W. Truelson
Registration No. 34,265

Telephone: (507) 289-6256

Docket No.: ROC9-1997-0187-US2
Serial No.: 09/866,981